

Amendments to the Specification:

Please replace paragraph [0009] with the following amended paragraph:

[0009] In various embodiments, the transformation process occurs in two steps. The first step includes identification of an appropriate transform process definition and addition of this transform process definition to the input data. The result of this step is a single data set including both the input data and part of the transform schema configured to direct the conversion of the data to be transformed to a desired output. In the second step, this single data set is processed by a transformation engine to generate the desired output.

Please replace paragraph [0010] with the following amended paragraph:

[0010] Various embodiments of the invention include a data transformation system comprising a data interface configured to receive data to be transformed or to send transformed data, memory configured to store one or more transform process definitions having at least one simple transform definition and at least one compound transform definition, an application including computer instructions, and a data interpreter configured to exchange data with the data interface and the application, the data interpreter including a transform engine configured to select a transform process definition from the one or more transform process ~~definition~~ definitions, the selected transform process definition including a hierarchical data structure, concurrently navigate the selected transform process definition and the data to be transformed, navigation within the data to be transformed being ~~response~~ responsive to transform definitions within the selected transform process definition, and generate output data having a data structure responsive to a data structure of the selected transform process definition.

Please replace paragraph [0011] with the following amended paragraph:

[0011] Various embodiments of the invention include a data interpreter configured to transform data to be transformed, the data interpreter comprising at least one computing device, and a transform engine supported by the computing device, the transform engine being configured to

access a transform process definition including a hierarchical data structure of transform definitions, the data structure including a simple transform definition and a compound transform definition, concurrently navigate the transform process definition and the data to be transformed, navigation within the data to be transformed being ~~response~~responsive to the transform definition within the transform process definition, and generate output data having a data structure responsive to the transform process definition.

Please replace paragraph [0040] with the following amended paragraph:

[0040] The bulk of the data included in Active XML 160 ~~have~~has been identified and selected from Transform Schema 170 for use in a particular transformation (*i.e.*, transformation of the data shown in FIGS. 3A-3F. FIG. 4A line 1 through ~~FIG. 4~~line 11 include identification and version data). This line identifies the start of a compound data element in which the transform process definition is embodied. Further data, retrieved from Transform Schema 170, based on identifying data found in the Meta-language Transform Input Data 200, starts at FIG. 4A line 12. This line identifies other characteristics of the transformation, such as that it is configured for "IA" (inbound asynchronous) data. Transform Schema 170 typically includes data defining more than one type of transformation that may be selected for inclusion in Active XML 160 based on the data type, data source, direction of data transmission, or the like. However, in a typical embodiment, only the transform process definition necessary to perform a particular transformation is selected from Transform Schema 170 for inclusion in Active XML 160. Active XML 160 may be stored temporarily as needed to perform a transformation.

Please replace paragraph [0075] with the following amended paragraph:

[0075] In various embodiments, transform definitions, included in Active XML 160, include further advanced functions configured to perform logic operations. For example, in some embodiments, value translation optionally includes a call to a function. The Active XML 160 illustrated in FIG. 4A-4B includes a function call at FIG. 4A line 55 and FIG. 4B line 56. The parameter "translation_codeset" is used to specify a function labeled COUNTRY_CODESET.

The definition of this function is typically found elsewhere in Active XML 160. In the illustrated example the definition is at FIG. 4B lines 80 through 87. The value of the element COUNTRY ("USA") is passed to this function from Payload Data 230 at FIG. 3F-3E lines 244 or 271. Using the computer instructions included in FIG. 4B lines 80 through 87, the value USA is transformed to "United States of America" and a new leaf element is created for STATE that holds the transformed value "California." The translation_codeset COUNTRY_CODESET includes a call to an external codeset PS_COUNTRY_01. (PS_COUNTRY_01 is, for example, a pointer to a function or a lookup table.) Through this call, or similar calls, Active XML 160 can invoke a variety of dynamic internal or external operations to process the data to be transformed.